operating parameter value, a mutation rate and a set of probabilistic mutation criteria are stored in a second portion of code. The method further comprises, responsive to an occurrence of a predetermined event associated with the environment, using the mutation rate and the probabilistic mutation criteria to determine if the operating parameter value should mutate, and changing the operating parameter value in response to a determination that the operating parameter value should mutate.

In the Claims:

Please delete claims 20-22.

Please add new claims 23-57 as follows:

--23. A method for adapting a software product to an environment, the method comprising:

providing the software product with a first portion of code including instructions for performing a predetermined function, at least one aspect of performing the predetermined function being controllable by an operating parameter value;

providing the software product with a second portion of code having stored therein the operating parameter value, a mutation rate and a set of probabilistic mutation criteria; and

responsive to an occurrence of a predetermined event associated with the environment,

using the mutation rate and the probabilistic mutation criteria to determine if the

operating parameter value should mutate, and

changing the operating parameter value in response to a determination that the

operating parameter value should mutate.--

24. A method according to claim 23 further comprising:

providing the software product with a third portion of code that includes instructions

for carrying out the step of using the mutation rate and the probabilistic

mutation criteria to determine if the operating parameter value should mutate

and the step of changing the operating parameter value.--

25. A method according to claim 23 wherein the step of changing the operating

parameter value includes substantially randomly selecting a new operating parameter

from a predetermined range of values.--

26. A method according to claim 23 wherein the predetermined event is selected from a

group consisting of a user-to-user copying event, operation of the product, unlocking of a

product feature and a change to a predetermined operating parameter of the product.--

27. A method according to claim 23 wherein the second portion of code has stored

therein a set of lineage-relevant information relating to a plurality of lineage-relevant

events resulting in the product, and wherein the method further comprises:

responsive to the occurrence of a predetermined event, modifying the set of lineage-

relevant information to include information relating to the occurrence of the

predetermined event.--

--28. A method according to claim 27 further comprising:

transmitting the set of lineage-relevant information to a central database.--

--29. A method according to claim 28 further comprising:

changing the mutation rate to a predetermined replacement mutation rate.--

--30. A method according to claim 29 wherein the desirable replacement mutation rate is

zero.--

--31. A method according to claim 30 further comprising:

changing the operating parameter value to a desirable fixed value.--

--32. A method according to claim 27 wherein the set of lineage-relevant information is

stored in the second portion of code as a plurality of bits, at least one of the bits having

been selected at random and its state changed in response to each of the plurality of

lineage-relevant events resulting in the product.--

--33. A method for adapting a software product to an environment, the method

comprising:

providing the software product with a first portion of code including instructions for

performing a predetermined function, at least one aspect of performing the

predetermined function being controllable by an operating parameter value;

providing the software product with a second portion of code having stored therein

the operating parameter value, a mutation rate, a set of probabilistic mutation

criteria and a set of lineage-relevant information relating to a plurality of

lineage-relevant events resulting in the product;

providing the software product with a third portion of code that includes instructions

for changing the operating parameter value in response to an occurrence of a

predetermined event associated with the environment and a determination

that the operating parameter value should mutate; and

responsive to an occurrence of a predetermined event associated with the

environment,

using the mutation rate and the probabilistic mutation criteria to determine if the

operating parameter value should mutate, and

changing the operating parameter value in response to a determination that the

operating parameter value should mutate.--

--84. A method according to claim 33 wherein the step of changing the operating

parameter value includes substantially randomly selecting a new operating parameter

from a predetermined range of values.--

31

--35. A method according to claim 33 wherein the predetermined event is selected from a

group consisting of a user-to-user copying event, operation of the product, unlocking of a

product feature and a change to a predetermined operating parameter of the product.--

--36. A method according to claim 33 further comprising:

   responsive to the occurrence of a predetermined event, modifying the set of lineage-

      relevant information to include information relating to the occurrence of the

      predetermined event.--

--37. A method according to claim 36 further comprising:

   transmitting the set of lineage-relevant information to a central database.--

--38. A method according to claim 37 further comprising:

   transmitting a replacement mutation rate from the central database to the software

      product.--

--39. A method according to claim 38 wherein the replacement mutation rate is zero.--

--40. A method according to claim 37 further comprising:

   transmitting a replacement operating parameter value from the central database to the

      software product.--

--41. A method according to claim 33 wherein the set of lineage-relevant information is

stored in the second portion of code as a plurality of bits, at least one of the bits having

32

been selected at random and its state changed in response to each of the plurality of

lineage-relevant events resulting in the product.--

20
--42. A method for adapting a software product to an environment, the method

comprising:

> providing the software product with a first portion of code including instructions for
>
>> performing a predetermined function, at least one aspect of performing the
>>
>> predetermined function being controllable by an operating parameter value;
>
> providing the software product with a second portion of code having stored therein
>
>> the operating parameter value, a mutation rate, a set of probabilistic mutation
>>
>> criteria and a set of lineage-relevant information relating to a plurality of
>>
>> lineage-relevant events resulting in the product;
>
> providing the software product with a third portion of code that includes instructions
>
>> for changing the operating parameter value in response to an occurrence of a
>>
>> predetermined event associated with the environment and a determination
>>
>> that the operating parameter value should mutate; and
>
> responsive to an occurrence of a predetermined event selected from a group
>
>> consisting of a user-to-user copying event, operation of the product,
>>
>> unlocking of a product feature and a change to a predetermined operating
>>
>> parameter of the product,

using the mutation rate and the probabilistic mutation criteria to determine if the

operating parameter value should mutate,

changing the operating parameter value by substantially randomly selecting a

new operating parameter from a predetermined range of values in

response to a determination that the operating parameter value should

mutate, and

modifying the set of lineage-relevant information to include information relating

to the occurrence of the predetermined event.--

--43. A method according to claim 42 further comprising:

transmitting the set of lineage-relevant information to a central database.--

--44. A method according to claim 43 further comprising:

transmitting a replacement mutation rate from the central database to the software

product.--

--45. A method according to claim 43 further comprising:

transmitting a replacement operating parameter value from the central database to

the software product.--

--46. A method according to claim 42 wherein the set of lineage-relevant information is

stored in the second portion of code as a plurality of bits, at least one of the bits having

34

been selected at random and its state changed in response to each of the plurality of

lineage-relevant events resulting in the product.--

--~~47~~. A method for adapting a software product to an environment, the method

comprising:

> providing a plurality of software products each having a first portion of code
>
>> including instructions for performing a predetermined function, at least one
>>
>> aspect of performing the predetermined function being controllable by an
>>
>> operating parameter value, a second portion of code having stored therein the
>>
>> operating parameter value, a mutation rate, a set of probabilistic mutation
>>
>> criteria and a set of lineage-relevant information relating to a plurality of
>>
>> lineage-relevant events resulting in the product, and a third portion of code
>>
>> that includes instructions for changing the operating parameter value in
>>
>> response to an occurrence of a predetermined event associated with the
>>
>> environment and a determination that the operating parameter value should
>>
>> mutate;
>
> receiving at a central database a transmission of the lineage-relevant information
>
>> from at least a portion of the plurality of software products; and
>
> performing a statistical analysis of the lineage-relevant information from all of the at
>
>> least a portion of the plurality of software products to ascertain a set of
>>
>> desirable characteristics for the software product.--

35

--48. A method according to claim 47 further comprising:

determining from the statistical analysis a desirable replacement mutation rate; and

transmitting the desirable replacement mutation rate to at least one of the plurality of

software products.--

--49. A method according to claim 47 further comprising:

determining from the statistical analysis a desirable replacement operating parameter

value; and

transmitting the desirable replacement operating parameter value to at least one of

the plurality of software products.--

--50. A computer program product for performing a predetermined function, the product

comprising:

a computer readable medium;

a first portion of code that is stored on the medium and that includes instructions for

performing the predetermined function, at least one aspect of performing the

predetermined function being controllable by an operating parameter value;

a second portion of code that is stored on the medium and that includes the operating

parameter value, a mutation rate and a set of probabilistic mutation criteria;

and

a third portion of code that is stored on the medium and that includes instructions for

modifying the operating parameter value in response to a combination of an

occurrence of a predetermined event and a determination that the

probabilistic mutation criteria have been met.--

--31. A computer program product according to claim 30 wherein the third portion of

code includes means for monitoring changes to the environment and means for

identifying the occurrence of a predetermined event.--

--32. A computer program product according to claim 30 wherein the predetermined

event is selected from a group consisting of a user-to-user copying event, operation of the

product, unlocking of a product feature and a change to a predetermined operating

parameter of the product.--

--33. A computer program product according to claim 30 wherein the second portion of

code includes a set of lineage-relevant information relating to at least one lineage-

relevant event resulting in the product and wherein the second portion of code is

configured for receiving additional information relating to at least one of a user-to-user

copying event, operation of the product, unlocking of features of the product and a

change to operating parameters of the product.--

--34. A computer program product according to claim 33 wherein the lineage-relevant

information is stored in the second portion of code as a plurality of bits, at least one of the

37

bits having been selected at random and its state changed in response to each lineage-

relevant event in the chain resulting in the product.--

--33. A computer program product for performing a predetermined function, the product

comprising:

a computer readable medium;

a first portion of code that is stored on the medium and that includes instructions for

performing the predetermined function, at least one aspect of performing the

predetermined function being controllable by an operating parameter value;

a second portion of code that is stored on the medium and that includes the operating

parameter value, a mutation rate, a set of probabilistic mutation criteria and a

set of lineage-relevant information relating to at least one lineage-relevant

event resulting in the product, the second portion of code being configured

for receiving additional information relating to at least one of a user-to-user

copying event, operation of the product, unlocking of features of the product

and a change to operating parameters of the product;

a third portion of code that is stored on the medium and that includes means for

identifying an occurrence of a predetermined event, means for determining

that the probabilistic mutation criteria have been met and means for

modifying the operating parameter value in response to a combination of an